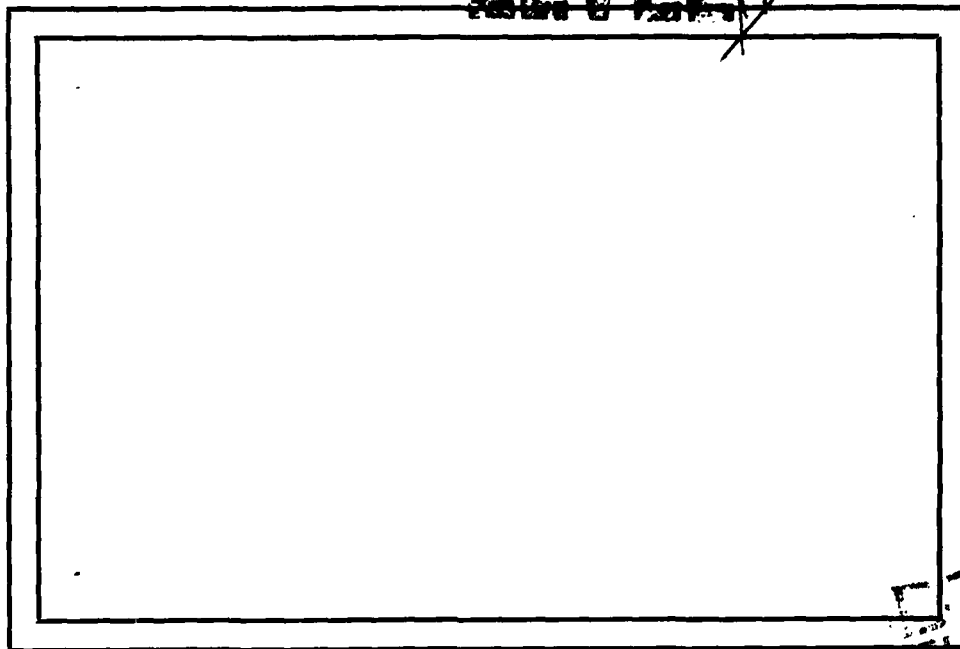


AD A090246

LEVEL

12



SELECTED
OCT 14 1980



UNIVERSITY OF MARYLAND
COMPUTER SCIENCE CENTER

COLLEGE PARK, MARYLAND

20742

DDC FILE COPY

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

80 10 3 114

①

(13) 241

14

TR-887
DAAG-53-76C-0138

11

April, 1980

6

GRAY-LEVEL CORNER DETECTION

10

Les/Kitchen
Azriel/Rosenfeld

Computer Vision Laboratory
Computer Science Center
University of Maryland
College Park, MD 20742

DIS
SELECT
OCT 14 1980

(15) DAAG 53-76-C-0138, DARPA Ord-3206

ABSTRACT

The usual approach to detecting corners in shapes involves first segmenting the shape, then locating the corners in its boundary. We present several techniques for detecting corners of shapes in gray-level images, without prior segmentation.

9 Technical Repts

The support of the Defense Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under Contract DAAG-53-76C-0138 (DARPA Order 3206) is gratefully acknowledged, as is the help of Kathryn Riley in preparing this paper.

Approved for Release
Distribution Unlimited

42-174

OK

Accession For	NTIS GRA&I ✓
DTIC TAB	Unannounced
Justification	
By	
Distribution/	
Availability	
Avail and	
Dist	Special

1. Introduction

The conspicuous corners of a shape play an important role in human perception [Zusne, 1970], and are of similar importance for the recognition of shapes by computer. Given a digital image of a shape, a typical approach to detecting its corners involves first segmenting the shape (by thresholding or some similar method), extracting its boundary as a chain code, and then searching for significant turnings in the boundary. Rutkowski and Rosenfeld [1978], and the references cited therein, provide a good survey of such techniques.

However, these techniques rely on prior segmentation of the shape, and will be led astray by errors in the segmentation. It is therefore of interest to develop techniques of corner detection which can be applied directly to a gray-level image, without the need for prior segmentation. Another advantage of such techniques is that the corners detected in the gray-level image can provide important clues about the shape, and guide a later segmentation process. For example, corner points could be used as joints for a polygonal approximation to a shape, without explicit extraction of the shape's boundary.

In the next section we present a number of techniques for gray-level corner detection, along with examples of their use.

2. Gray-level corner detection techniques

Every technique discussed below involves the application of a local operator, in parallel, to neighborhoods of a gray-level picture. In each neighborhood, the operator computes some measure of curvature for an edge that passes through that neighborhood. However, such a measure will have high values, not only at corners of a shape, but spread over the image, because of noise and digitization artifacts. This can be remedied by multiplying the curvature measure by the local gradient magnitude. The resulting quantity will take high values only where there is a strong edge which turns rapidly, that is, at conspicuous corners. (For display, the absolute value of this quantity is further multiplied by a variable scale factor in order to make use of the full range of displayable gray levels.) The local gradient is calculated by applying horizontal and vertical Prewitt operators [Prewitt 1970] to measure the x and y components of the gradient, and then converting to polar coordinates in order to obtain gradient magnitude and direction. Two images are used as examples (see Figure 1). The first is a picture of a maple leaf with sharp edges and corners. The second is photomicrograph of chromosomes with blurred edges and ill-defined corners.

2.1 Gradient magnitude of gradient direction

The first method considered is based on the following observation: If gradient directions are taken as lying in the range -180 to $+180$ degrees, and their absolute values displayed as a gray-level picture, then this direction picture will show changes of brightness precisely where the original picture had changes of edge direction. (The choice of range and taking of absolute values are required to prevent a spurious discontinuity in edge directions at 180 degrees.) These brightness changes can be found by measuring the gradient magnitude of the direction picture. As indicated above, it will be necessary to multiply this result by the gradient magnitude of the original picture in order to obtain a true corner-detection measure.

Figure 2 shows some results obtained with this technique.

2.2 Change of direction along edge

The results obtained above, while they show promise, are rather disappointing, since the second application of the edge detector measures indiscriminately any change in edge direction. It is preferable to measure only direction changes along the edge, since these correspond more closely to turns in the boundary of an object. This can be achieved by once again measuring the gradient direction in the original picture, and then applying a 3 by 3 operator to the resulting picture. This operator examines each neighborhood, and determines which opposing pair of non-central pixels lie closest to the line that passes through the center pixel and is perpendicular to the gradient direction at the center pixel. The result of the operator is the difference between the gradient directions at the two pixels thus determined. If the signed difference is taken, and proper conventions are observed, it is possible to extract information about the direction of curvature, as well as its magnitude. Once again, this curvature measure must be multiplied by the gradient magnitude.

Some results obtained by this technique are shown in Figure 3. They are significantly better than those of method 2.1 above. It should be remarked at this point that both this and the previous technique effectively use 5 by 5 operators, since they involve the application of a 3 by 3 operator to the results of another 3 by 3 operator.

2.3 Angle between most similar neighbors

If an edge passes through the center of a neighborhood, then those pixels that lie along the edge should have gray levels similar to that of the center pixel, while those off the edge should be brighter or darker. In a 3 by 3 neighborhood, we can take the two non-central pixels nearest in gray level to the center pixel (call them A and B, and the center pixel C). We can then take the difference in direction between the vectors AC and CB, and use this difference as a measure of curvature.

Figure 4 shows some results obtained with this technique. They are hardly satisfactory, mostly because the 3 by 3 neighborhood used permits only four different changes of direction (0, 45, 90, and 135 degrees), and is unduly sensitive to noise. The method to be described next was designed to reduce this noise sensitivity.

2.4 Turning of fitted surface

In general, a property of a gray-level image can be computed by fitting a function of two spatial variables to the gray-level values in the image, and then determining the corresponding property of the fitted function by analytic means. Typically, the function is a polynomial of fairly low degree, which fits the gray-level data in a small local neighborhood with minimal sum of squared errors [Prewitt 1970, Beaudet 1978].

Suppose we have fitted a function $g(x,y)$ to the gray levels in a picture neighborhood. For simplicity, assume that the neighborhood is square, an odd number of pixels along a side, with the origin of a local Cartesian coordinate system at its center pixel. Let $\theta(x,y)$ be the gradient direction given by

$$\tan\theta = g_y/g_x$$

at any point (x,y) . (Below, the arguments of functions will be omitted for brevity. They are always assumed to be (x,y) .) The partial derivatives of θ are

$$\theta_x = \frac{g_{xy}g_x - g_{xx}g_y}{g_x^2 + g_y^2}$$

$$\theta_y = \frac{g_{yy}g_x - g_{xy}g_y}{g_x^2 + g_y^2}$$

Now, the gradient vector

$$(g_x, g_y)$$

is directed across the edge, so the vector

$$(-g_y, g_x)$$

(at right angles to the gradient) is directed along the edge. Projecting the change of gradient direction vector

$$(\theta_x, \theta_y)$$

along the edge, and multiplying the result by the local gradient magnitude, gives the result

$$\begin{aligned} k &= \frac{g_x \theta_y - g_y \theta_x}{g_x^2 + g_y^2} \\ &= \frac{g_{xx} g_y^2 + g_{yy} g_x^2 - 2g_{xy} g_x g_y}{g_x^2 + g_y^2} \end{aligned}$$

This quantity, evaluated at the center of the neighborhood, measures the rate of change of gradient direction along an edge, multiplied as usual by the gradient magnitude. It can be regarded as a continuous analog of method 2.2 above.

This same quantity can be derived in another way as follows: Consider the contour line passing through the center of the neighborhood, given by the equation

$$G(x,y) = g(x,y) - g(0,0) = 0$$

Assuming proper behavior of the function g , we can without loss of generality take this equation to define y as a function of x near the origin. By implicit differentiation we can determine the first and second derivatives of y with respect to x , in terms of the partial derivatives of g . By substituting into the expression for the curvature of a plane curve,

$$\frac{d^2y/dx^2}{(1 + (dy/dx)^2)^{3/2}}$$

and multiplying by the gradient magnitude, we again obtain the expression for k derived above. Thus k can also be regarded as the curvature of a contour line, the continuous analog of method 2.3 above.

Figures 5, 6, and 7 show results obtained using this method, fitting a second order polynomial surface to square neighborhoods of sizes 3, 5, and 7, respectively. The derivation of the best-fit surface is presented in the Appendix.

2.5 Beaudet's DET

Beaudet [1978] defines an operator called DET,

$$g_{xx}g_{yy} - g_{xy}^2$$

which responds at corners and saddle points of a surface. Figures 8, 9, and 10 show some results obtained with this operator, again using a second order polynomial surface and neighborhoods of sizes 3, 5, and 7, respectively. Notice that DET need not be multiplied by the gradient magnitude in order to produce meaningful results. In fact, DET does not respond at all when positioned exactly on an edge. Near a corner of a shape DET responds (with opposite signs) on both sides of the edge. DET fares badly with very sharp edges such as are found on the maple leaf.

2.6 Better localization of corners

The responses of most of the above detectors are somewhat spread out, especially with the larger sized neighborhoods. The localization of corners can be improved by pointwise multiplication of the outputs of detectors of various sizes (with appropriate rescaling), or by applying isotropic non-maximum suppression. (See Figure 11.)

A further problem arises with the methods that use multiplication by the gradient magnitude: If the edge near a corner is blurred, then the corner detector will respond all the way across the edge. This can be remedied by applying non-maximum suppression (along the gradient direction) to the edge magnitudes before using them for multiplication. Results of this improvement are shown in Figure 12.

3. Conclusions

We have investigated a number of techniques for detecting the corners of a shape in a gray-level image, prior to extraction of the shape. Of these techniques, the most successful appears to be method 2.4. Its success can be attributed to three causes: Firstly, the fitting of a surface virtually eliminates the effects of noise. Secondly, the fitted surface is of high enough order to capture the interesting properties of the neighborhood's gray-level pattern. For example, a fitted plane would not have been adequate. And thirdly, the operator used is based upon two intuitive characterizations of a gray-level corner which correspond closely to the notion of a corner in an already segmented shape. The results of method 2.5 (Beaudet's DET) are equally good, except for its failure at very sharp corners. It too is based on analytically derived properties of a fitted surface of adequate order. Various techniques can be used to improve the localization of corners detected by these methods.

Appendix: Best-fit surface

Suppose we have a subset of a digital picture given by the pixels with Cartesian coordinates (x_1, y_1) , (x_2, y_2) , ..., (x_n, y_n) with corresponding gray levels z_1, z_2, \dots, z_n . For a second order polynomial surface

$$g(x, y) = ax^2 + by^2 + cxy + dx + ey + f$$

the sum of squared errors on this subset is given by

$$\sum_{i=1}^n (z_i - (ax_i^2 + by_i^2 + cx_i y_i + dx_i + ey_i + f))^2$$

The quadratic surface which minimizes this error can be found by expanding this expression, then differentiating partially with respect to the six parameters a , b , c , d , e , and f . At the minimum, these partial derivatives must all be zero, leading to a set of six simultaneous linear equations in the parameters, with coefficients of the form

$$\sum_{i=1}^n x_i^p y_i^q z_i^r$$

which we will denote by the shorthand notation S_{pqr} .

If we restrict our attention to a square subset of a picture, an odd number of pixels along a side, with the origin at the center pixel of the square, then the system of equations is considerably simplified. In particular,

$$S_{pq0} = S_{qp0}$$

and if p or q is odd, then

$$S_{pq0} = 0$$

Making use of the additional identity that

$$s_{200}^2 = s_{000}s_{220}$$

(easily proven by induction on the side of the neighborhood), we can fairly readily derive the following expressions for the parameters of the best-fit quadratic. Let

$$\alpha = \frac{1}{s_{400} - s_{220}}$$

$$\beta = \frac{s_{200}}{s_{000}s_{400} - s_{200}^2}$$

$$\gamma = \frac{s_{000}s_{400} + s_{200}^2}{s_{000}(s_{000}s_{400} - s_{200}^2)}$$

Then

$$a = \alpha s_{201} - \beta s_{001}$$

$$b = \alpha s_{021} - \beta s_{001}$$

$$c = \frac{s_{111}}{s_{220}} \quad d = \frac{s_{101}}{s_{200}} \quad e = \frac{s_{011}}{s_{200}}$$

$$f = \gamma s_{001} - \beta (s_{201} + s_{021})$$

For the 3 by 3 case, $\alpha = 1/2$, $\beta = 1/3$, $\gamma = 5/9$,

$$s_{220} = 4, \text{ and } s_{200} = 6.$$

Relating these coefficients to Beaudet's notation, we have

$$I = g(0,0) = f$$

$$I_x = g_x(0,0) = d$$

$$I_y = g_y(0,0) = e$$

$$I_{xy} = g_{xy}(0,0) = c$$

$$I_{xx} = g_{xx}(0,0) = 2a$$

$$I_{yy} = g_{yy}(0,0) = 2b.$$

These derivatives can be computed directly from an image by the application of appropriate linear templates. See Beaudet's paper for these templates, and templates for higher order derivatives based on higher order fitted polynomials. (Note that he uses row-column coordinates rather than Cartesian coordinates.) Morgenthaler and Rosenfeld [1980] derive first partial derivatives for "images" with more than two dimensions.

References

- P. R. Beaudet, "Rotationally invariant image operators", International Joint Conference on Pattern Recognition, 1978, pp. 579-583.
- D. G. Morgenthaler and A. Rosenfeld, "Multidimensional edge detection by hypersurface fitting", Technical Report 877, Computer Science Center, University of Maryland, College Park, MD 20742, February 1980.
- J. M. S. Prewitt, "Object enhancement and extraction", in Lipkin and Rosenfeld (Eds.), Picture Processing and Psychopictorics, Academic Press, New York, 1970, pp. 75-149.
- W. S. Rutkowski and A. Rosenfeld, "A comparison of corner-detection techniques for chain-coded curves", Technical Report 623, Computer Science Center, University of Maryland, College Park, MD 20742, January 1978.
- L. Zusne, Visual Perception of Form, Academic Press, New York, 1970, p. 59f.

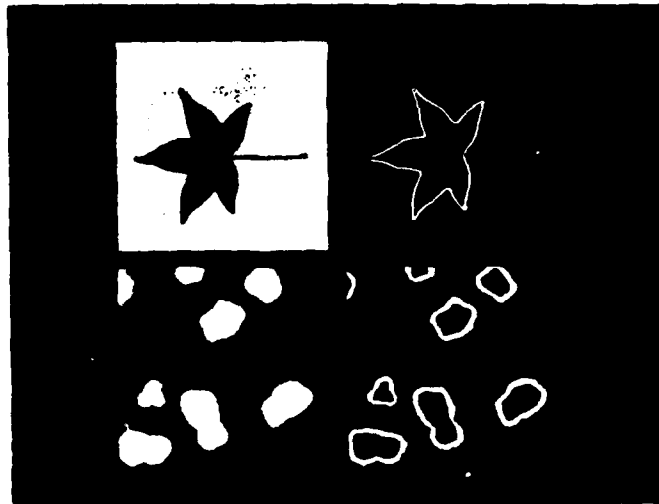


Figure 1

- (a) (Top-left) Picture of maple leaf
- (b) (Top-right) Gradient magnitude of 1a (3x3 Prewitt)
- (c) (Bottom-left) Photomicrograph of chromosomes
- (d) (Bottom-right) Gradient magnitude of 1c (3x3 Prewitt)



Figure 2

- (a) (Top-left) Method 2.1 applied to 1a (scale factor 7.0)
- (b) (Top-right) Figure 2a multiplied by gradient magnitude (scale factor 0.412)
- (c) (Bottom-left) Method 2.1 applied to 1c (scale factor 7.0)
- (d) (Bottom-right) Figure 2c multiplied by gradient magnitude (scale factor 0.875)

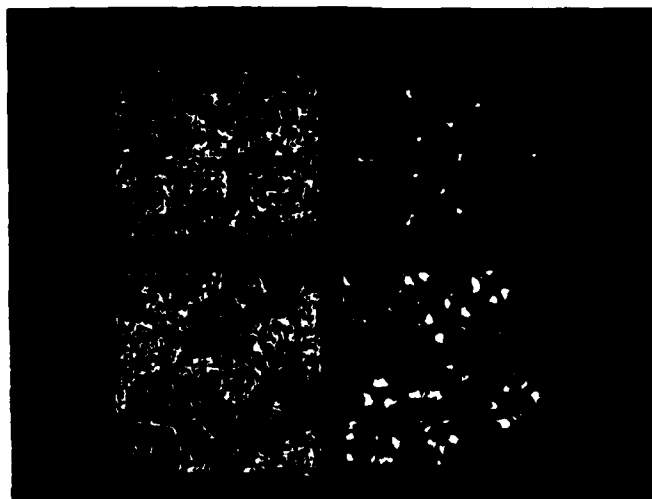


Figure 3

- (a) (Top-left) Method 2.2 applied to 1a (scale factor 20.0)
- (b) (Top-right) Figure 3a multiplied by gradient magnitude (scale factor 1.18)
- (c) (Bottom-left) Method 2.2 applied to 2c (scale factor 20.0)
- (d) (Bottom-right) Figure 3c multiplied by gradient magnitude (scale factor 2.5)

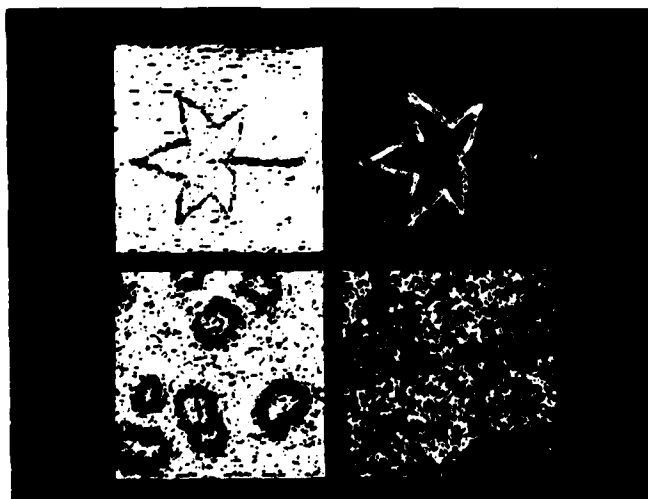


Figure 4

- (a) (Top-left) Method 2.3 applied to 1a (scale factor 21.0)
- (b) (Top-right) Figure 4a multiplied by gradient magnitude (scale factor 1.24)
- (c) (Bottom-left) Method 2.3 applied to 1c (scale factor 21.0)
- (d) (Bottom-right) Figure 4c multiplied by gradient magnitude (scale factor 2.63)



Figure 5 (Method 2.4 using 3x3 neighborhoods)

- (a) (Top-left) Method 2.4 applied to 1a (scale factor 10.0)
- (b) (Top-right) Figure 5a multiplied by gradient magnitude (scale factor 10.0)
- (c) (Bottom-left) Method 2.4 applied to 1c (scale factor 20.0)
- (d) (Bottom-right) Figure 5c multiplied by gradient magnitude (scale factor 30.0)

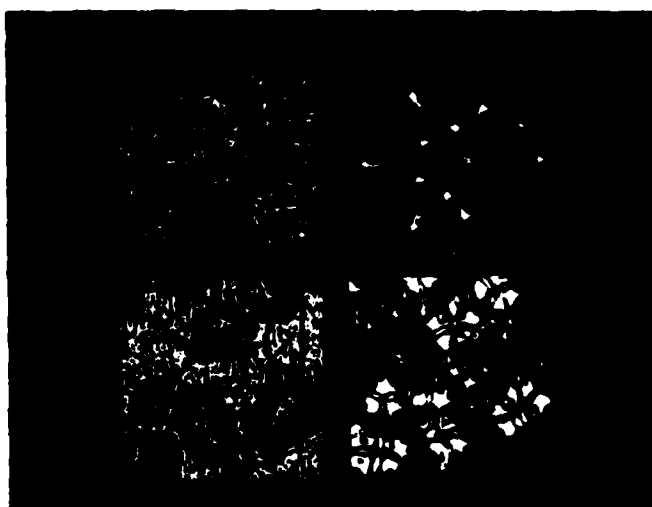


Figure 6 (Analogous to Figure 5, but using 5x5 neighborhoods)

- (a) (Top-left) scale factor 20.0
- (b) (Top-right) scale factor 20.0
- (c) (Bottom-left) scale factor 40.0
- (d) (Bottom-right) scale factor 60.0

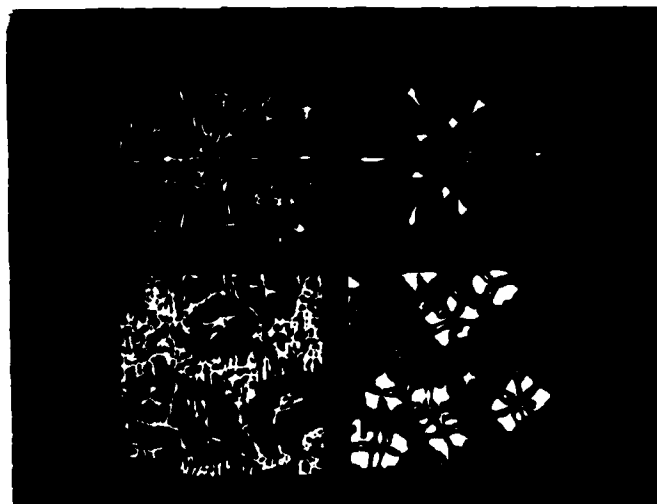


Figure 7 (Analogous to Figure 5, but using 7x7 neighborhoods)

- (a) (Top-left) scale factor 40.0
- (b) (Top-right) scale factor 25.0
- (c) (Bottom-left) scale factor 80.0
- (d) (Bottom-right) scale factor 80.0



Figure 8 (Method 2.5 using 3x3 neighborhoods)

- (a) (Top-left) Method 2.5 applied to 1a (scale factor 2.0)
- (b) (Top-right) Figure 8a multiplied by gradient magnitude (scale factor 0.1)
- (c) (Bottom-left) Method 2.5 applied to 1c (scale factor 20.0)
- (d) (Bottom-right) Figure 8c multiplied by gradient magnitude (scale factor 2.5)

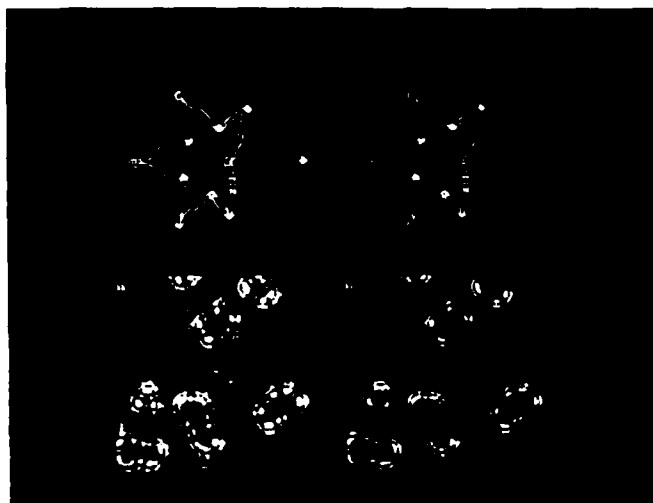


Figure 9 (Analogous to Figure 8, but using 5x5 neighborhoods)

- (a) (Top-left) scale factor 10.0
- (b) (Top-right) scale factor 1.0
- (c) (Bottom-left) scale factor 40.0
- (d) (Bottom-right) scale factor 5.0

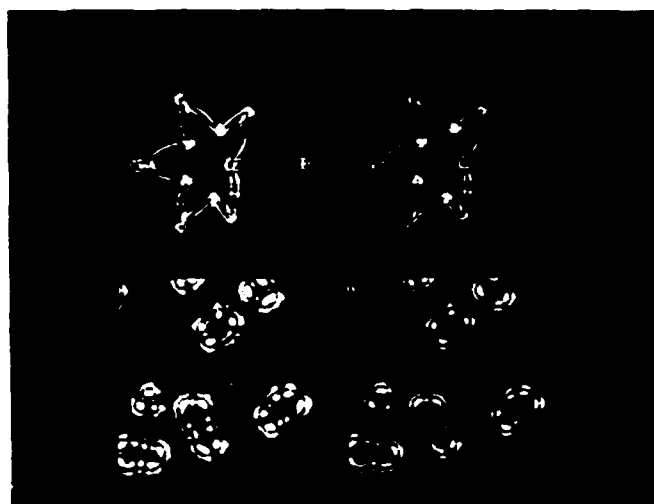


Figure 10 (Analogous to Figure 8, but using 7x7 neighborhoods)

- (a) (Top-left) scale factor 40.0
- (b) (Top-right) scale factor 4.0
- (c) (Bottom-left) scale factor 80.0
- (d) (Bottom-right) scale factor 10.0



Figure 11

- (a) (Top-left) Figures 5b, 6c, and 7b multiplied together
- (b) (Top-right) Isotropic non-maximum suppression applied to Figure 7b
- (c) (Bottom-left) Figures 5d, 6d, and 7d multiplied together
- (d) (Bottom-right) Isotropic non-maximum suppression applied to Figure 7d



Figure 12 (Uses gradient computed on 5x5 neighborhood)

- (a) (Top-left) Gradient magnitude of 1a after non-maximum suppression along gradient direction (scale factor 5.0)
- (b) (Top-right) Same as 6b, but using gradient magnitude after non-maximum suppression along gradient direction
- (c) (Bottom-left) Gradient magnitude of 1c after non-maximum suppression along gradient direction (scale factor 5.0)
- (d) (Bottom-right) Same as 6d, but using gradient magnitude after non-maximum suppression along gradient direction

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A090 146	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) GRAY-LEVEL CORNER DETECTION		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER TR-887
7. AUTHOR(s) Les Kitchen Azriel Rosenfeld		8. CONTRACT OR GRANT NUMBER(s) DAAG-53-76C-0138
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD 20742		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Night Vision Laboratory Ft. Belvoir, VA 22060		12. REPORT DATE April 1980
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Image processing Pattern recognition Feature detection Corners Angles		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The usual approach to detecting corners in shapes involves first segmenting the shape, then locating the corners in its boundary. We present several techniques for detecting corners of shapes in gray-level images, without prior segmentation.		

DD FORM 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)